



Fakultät Informatik

Studiengang Informatik (Master)

Modulhandbuch

Stand: WS 2026/27

Juli 2026

Prof. Dr. E. Müller
Studiendekan der Fakultät Informatik

Prof. Dr. E. Böhler
Studiengangkoordinator

Prof. Dr. U. Göhner
Vorsitzender der Prüfungskommission

Inhaltsverzeichnis

1 Ziele und Aufbau des Studiengangs Informatik	2
2 Begriffserläuterungen	8
3 Modulbeschreibungen	10
IFM1101 Effiziente Algorithmen	10
IFM1102 Mathematik	12
IFM1103 Softwarearchitektur	14
IFM5101 3D Computer Vision	16
IFM5102 Deep Learning für Computer Vision	18
IFM5103 Large Language Models	20
IFM5104 Reinforcement Learning	22
IFM5105 Physical AI	24
IFM5301 Betriebssystem im Eigenbau	26
IFM5302 Echtzeitsysteme	27
IFM5303 Multimodale Sensorsysteme	29
IFM5304 Mobile Roboter	31
IFM5309 Kryptographie	33
IFM5311 Computer Vision	35
IFM5402 Identity- und IT-Architekturmanagement	37
IFM5405 Integration und Management von Geschäftsprozessen mit SAP	39
IFM5406 Funktionale Sicherheit	41
IFM5407 E-Business Management	43
IFM5409 Requirements Engineering und Management	45
IFM5601 Deep Learning	47
IFM5603 Datenvisualisierung	49
IFM5604 Parallelprogrammierung	51
IFM5605 Algorithmen und Strategien zur Entscheidungsunterstützung	53
IFM5606 Big Data	55
IFM5609 Data Science	57
IFM2000 Hauptseminar	59
IFM3000 Projekt	60
IFM6000 Masterarbeit	61

1 Ziele und Aufbau des Studiengangs Informatik

Ziele des Masterstudiengangs Informatik sind die Vermittlung verschiedenster Kompetenzen und Lehrinhalte. Der Studiengang ist als anwendungsorientierter postgradualer konsekutiver Studiengang konzipiert. Er baut inhaltlich auf den Bachelor- oder Diplomstudiengängen Informatik, Wirtschaftsinformatik und Informatik- Game Engineering auf.

Der Studiengang ist technologieorientiert und deckt zentrale Gebiete der Informatik ab. Er qualifiziert die Studierenden für das Tätigkeitsfeld Informatik sowie für anwendungsorientierte Forschung und legt dabei besonderen Wert auf die theoretisch-wissenschaftlichen Grundlagen.

In der folgenden Tabelle werden den übergeordneten abstrakten Kompetenzen (Studienziele) die jeweiligen Lernergebnisse zugeordnet:

Nr.	Studienziele	Lernergebnisse
1	Formale, algorithmische und mathematische Kompetenz	Fundierte Wissen über die Grundlagen der Informatik und der Mathematik Fähigkeit das erworbene Wissen in der Praxis anzuwenden und selbständig auszuweiten
2	Analyse-, Design- und Realisierungs-Kompetenz	Fähigkeit zur Lösung von ungewöhnlichen und Nicht-Standard-problemen Fähigkeit aus unvollständigen und widersprüchlichen Informationen in Abstimmung mit dem Auftraggeber konsistente und überzeugende Lösungen zu erarbeiten und umzusetzen
3	Technologische Kompetenz	Vertiefte Kenntnis der technologischen Grundlagen mindestens eines Anwendungsbereichs der Informatik Fähigkeit sich in neue Technologien einzuarbeiten, Technologien auszuwählen und einzusetzen Fähigkeit Technologien wissenschaftlich fundiert weiterzuentwickeln
4	Methodenkompetenz	Fähigkeit das erworbene Wissen auch in komplexen Anwendungsszenarien einzusetzen und erfolgreich anzuwenden Befähigung zu wissenschaftlicher Arbeit, Gute wissenschaftliche Praxis
5	Projektmanagementkompetenz	Fähigkeit auch größere Projekte zu planen, zu organisieren und zu leiten, sowie gegenüber Dritten zu vertreten Probleme frühzeitig zu erkennen und zu lösen und Projekte erfolgreich abzuschließen
6	Anwendungskompetenz	Umfassendes fachliches und technisches Wissen und vertiefte Kenntnis der Vorgehensweisen und Prozesse in einem oder mehreren Anwendungsbereichen der Informatik Verständnis für die rechtlichen und nichttechnischen Auswirkungen von Anwendungen der Informatik

Der Zusammenhang zwischen den übergeordneten Zielen und den Lernergebnissen des Masterstudiengangs sowie der Beitrag der Pflicht- und Wahlpflichtmodule zur Umsetzung dieser Ziele sind in der folgenden Zielematrix dargestellt:

Modul	Studienziel					
	1	2	3	4	5	6
3D Computer Vision	++	+	++	++		+
Algorithmen und Strategien zur Entscheidungsunterstützung	+		+	++		+
Augmented Reality			+			++
Betriebssystem im Eigenbau			++			
BigData	+	+	+			++
Computer Vision	++		+			++
Data Science	+	++		+		+
Datenschutz				+	+	++
Datenvisualisierung	+		++			++
Deep Learning	+		+			++
Deep Learning für Computer Vision	+	+	++	++		++
E-Business-Management		+	+	+		++
Echtzeitsysteme			+			++
Effiziente Algorithmen	++		+			
eHealth	+	+	+			++
Funktionale Sicherheit		+	+	+	++	++
Hauptseminar			+	++		
Identity- und IT-Architektur-Management		+		++	++	+
Integration und Management von Geschäftsprozessen mit SAP		+	+	+	++	++
Kryptographie	++	+	+			
Large Language Models	+		++	+		+
Masterarbeit	+	++	+	++		++
Mathematik	++	+				
Mobile Roboter	+		+			++
Multimodale Sensorsysteme			+			++
Mustererkennung und maschinelles Lernen	+		+			++
Parallelprogrammierung			++			
Projekt	+	+	+	++	++	+
Reinforcement Learning	++	+	++	+		+
Requirements Engineering und Management		++		+	+	
Softwarearchitektur		++	+	+		

Absolventen des Masterstudiengangs Informatik erwerben während ihres Studiums eine Zusatzqualifikation, durch die sie mittels wissenschaftlicher Methodik anspruchsvolle Aufgaben der Informatik bearbeiten können. Während die Absolventen im Bereich der Grundlagen der Informatik und Mathematik ein hervorragendes Fundament erworben haben, befähigt der Masterstudiengang sie das große Potenzial auf Anwendungsseite zu beherrschen. Die Komplexität von Anwendungen steigt unaufhörlich und erfordert eine entsprechende Qualifikation. Die Absolventen sollen aufbauend auf ihren erworbenen fachlichen, methodischen und anwendungsspezifischen Kompetenzen in der Lage sein, komplexe Anwendungsszenarien zu verstehen, problemadäquate Technologien auszuwählen und einzusetzen, zu Problemen auch bei wider-

sprüchlicher und unvollständiger Information Lösungskonzepte zu entwickeln und zu vertreten, Projekte entsprechend zu planen, zu leiten und durchzuführen.

Zielgruppe des Masterstudiengangs sind Absolventen der Informatik, der Wirtschaftsinformatik, Informatik-Game Engineering oder eines gleichwertigen Studiengangs an einer Hochschule.

Das Konzept des Masterstudiengangs in Voll- und Teilzeit beruht auf dem soliden Fundament der drei Pflichtmodule Effiziente Algorithmen, Mathematik und Softwarearchitektur. Auf diesem Fundament bauen die Wahlpflichtmodule auf. Die Wahlpflichtmodule sind den drei Anwendungsbereichen "Autonome Systeme und Technische Informatik", "Wirtschaftsinformatik und E-Business" und "Data Science" zugeordnet. Bei diesen Wahlpflichtmodulen besteht für die Studierenden eine freie Kombinierbarkeit, welche diesen sowohl eine vertiefte Spezialisierung, als auch den Erwerb eines breiten Fachwissens ermöglicht. In den Pflichtveranstaltungen Hauptseminar und Projektarbeit beweisen und vertiefen die Studierenden sowohl anwendungsorientiert-wissenschaftliche als auch methodische- und Teamkompetenz.

Qualifikationsvoraussetzung für den Masterstudiengang in Vollzeit ist ein mindestens mit der Note 2,5 abgeschlossenes Studium der Informatik, der Wirtschaftsinformatik, Informatik-Game Engineering oder eines gleichwertigen Abschlusses. Wurden in einem Bachelorstudiengang weniger als 210 ECTS, mindestens aber 180 ECTS erworben, so sind die bis zum Kompetenzniveau von 300 ECTS fehlenden ECTS-Punkte aus den grundständigen Studiengängen der Hochschule Kempten nachzuerbringen. Welche Module noch zu erbringen sind bestimmt die Prüfungskommission.

Über die Gleichwertigkeit der Abschlüsse entscheidet die Prüfungskommission. Ist ein Abschluss nicht gleichwertig, kann die Gleichwertigkeit des Abschlusses durch Entscheidung der Prüfungskommission auch dadurch erzielt werden, dass fehlende Kompetenzen im Umfang von max. 30 ECTS-Punkten aus den grundständigen Studiengängen der Hochschule Kempten erworben werden.

Voraussetzung für die Teilnahme am Studiengang in Teilzeit ist das Erfüllen derselben Qualifikationsvoraussetzungen wie im Vollzeitstudiengang.

Aufbau des Studiengangs Informatik (Master)

Studiengang in Vollzeit

Informatik, Master (M.Sc.)

Hochschule für angewandte Wissenschaften Kempten

gültig mit Studienbeginn WS 2017/18

Semester

3	Masterarbeit ⁽⁶⁾																																																												
2	Projekt ⁽⁴⁾					WP-Fach ⁽⁵⁾					WP-Fach ⁽⁵⁾					WP-Fach ⁽⁵⁾					WP-Fach ⁽⁵⁾					WP-Fach ⁽⁵⁾																																			
1	Effiziente Algorithmen ⁽³⁾					Mathematik ⁽¹⁾					Softwarearchitektur ⁽²⁾					Hauptseminar ⁽⁴⁾					WP-Fach ⁽⁵⁾					WP-Fach ⁽⁵⁾																																			
<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td><td>26</td><td>27</td><td>28</td><td>29</td><td>30</td></tr></table>																															1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	ECTS-Punkte
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30																																

Legende:

1	Mathematik	4	Angewandte Informatik und interdisziplinäre Fächer
2	Softwaretechnik	5	Wahlpflichtmodul
3	Theorie der Informatik	6	Seminar- und Abschlussarbeit

Zuordnung der Wahlpflichtmodule zu Anwendungsbereichen

	1)	2)	3)
3D Computer Vision	X		X
Algorithmen und Strategien zur Entscheidungsunterstützung	X	X	
Augmented Reality			X
Betriebssysteme im Eigenbau			X
Big Data	X	X	
Computer Vision			X
Data Science	X		
Datenschutz		X	
Datenvisualisierung	X		
Deep Learning	X		X
Deep Learning für Computer Vision			X
E-Business-Management	X	X	
Echtzeitsysteme			X
eHealth		X	
Funktionale Sicherheit			X
Identity und IT-Architekturmanagement		X	
Integration und Management von Geschäftsprozessen mit SAP		X	
IPR (Intellectual Property Rights)		X	
Kryptographie			X
Large Language Models	X		
Mobile Roboter			X
Multimodale Sensorsysteme			X
Mustererkennung und maschinelles Lernen	X		X
Parallelprogrammierung	X		X
Physical AI			X
Reinforcement Learning			X
Requirements Engineering und Management		X	

1) Data Science

2) Wirtschaftsinformatik und E-Business

3) Autonome Systeme und Technische Informatik

Studiengang in Teilzeit

Informatik, Master (M.Sc.) - Teilzeit Hochschule für angewandte Wissenschaften Kempten

Semester	gültig mit Studienbeginn WS 2017/18														
6	Masterarbeit ⁽⁶⁾														
5	Masterarbeit ⁽⁶⁾														
4	Projekt ⁽⁴⁾					WP-Fach ⁽⁵⁾					WP-Fach ⁽⁵⁾				
3	Hauptseminar ⁽⁴⁾					WP-Fach ⁽⁵⁾					WP-Fach ⁽⁵⁾				
2	WP-Fach ⁽⁵⁾					WP-Fach ⁽⁵⁾					WP-Fach ⁽⁵⁾				
1	Effiziente Algorithmen ⁽³⁾					Mathematik ⁽¹⁾					Softwarearchitektur ⁽²⁾				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

ECTS-Punkte

1

Mathematik

2

Softwaretechnik

3

Theorie der Informatik

4

Angewandte Informatik und interdisziplinäre Fächer

5

Wahlfach

6

Seminar- und Abschlussarbeit

Für den Studiengang Informatik (Master) stehen folgende Ansprechpartner zur Verfügung:

Studiengangkoordinator:	Prof. Dr. Elmar Böhler
Fachstudienberater:	Prof. Dr. Jochen Staudacher
Vorsitzender der Prüfungskommission:	Prof. Dr. Ulrich Göhner

2 Begriffserläuterungen

ECTS - European Credit Transfer System

Diese Vereinbarungen zur Anrechnung, Übertragung und Akkumulierung von Studienleistungen basieren auf dem Arbeitspensum, das Studierende durchzuführen haben, um die Ziele des Lernprogramms zu erreichen. Für jede studienbezogene Leistung wird der voraussichtliche durchschnittliche Arbeitsaufwand angesetzt und auf das Studienvolumen angerechnet. Der Arbeitsaufwand umfasst Präsenzzeit und Selbststudium ebenso wie die Zeit für die Prüfungsleistungen, die notwendig sind, um die Ziele des vorher definierten Lernprogramms zu erreichen. Mit dem ECTS können Studienleistungen international angerechnet und übertragen werden.

Arbeitsaufwand (Workload) und Leistungspunkte (ECTS-LP)

Der Arbeitsaufwand der Studierenden wird im ECTS in credit points angegeben. Deutsche Übersetzungen für credit point sind die Begriffe Leistungspunkt oder ECTS-Punkt. Ein Arbeitsaufwand von 30 Zeitstunden bedeutet einen Leistungspunkt. Der Arbeitsaufwand von Vollzeitstudierenden entspricht 60 Leistungspunkten pro Studienjahr, also 30 Leistungspunkten pro Semester. Das sind 1.800 Stunden pro Jahr oder 45 Wochen/Jahr mit 40 Stunden/Woche.

Der Arbeitsaufwand setzt sich zusammen aus:

- Präsenzzeit
- Zeit für die Vor- und Nachbereitung des Vorlesungsstoffs,
- Zeit für die Vorbereitung von Vorträgen und Präsentationen,
- Zeit für die Erstellung eines Projekts,
- Zeit für die Ausarbeitung einer Studienarbeit,
- Zeit für notwendiges Selbststudium,
- Zeit für die Vorbereitung auf mündliche oder schriftliche Prüfungen.

Die Bachelorstudiengänge mit sieben Semestern bescheinigen erfolgreichen Studierenden 210 ECTS-LP, die dreisemestrigen Masterstudiengänge weitere 90 ECTS-LP. Damit ist die Forderung nach 300 ECTS-LP für ein erfolgreich abgeschlossenes Masterstudium erfüllt.

Semesterwochenstunden und Präsenzzeit

Eine Semesterwochenstunde ist die periodisch wiederkehrende Lehreinheit in einem Modul, in der Regel im Rhythmus von einer oder zwei Wochen. Dabei wird eine Präsenz von 45 Minuten plus Wegzeiten gerechnet, sodass die Vorlesungsstunde als eine Zeitstunde gewertet wird.

Wir rechnen mit einer Vorlesungszeit von 15 Wochen pro Semester, wodurch sich aus der Zahl der Semesterwochenstunden die geforderte Präsenzzeit ("Kontaktzeit") direkt ableitet: 1 SWS entspricht 15 Stunden Präsenzzeit.

Module und Wahlpflichtmodule

Der Studiengang setzt sich aus Modulen zusammen. Ein Modul repräsentiert eine inhaltlich und zeitlich zusammengehörige Lehr- und Lerneinheit. Module werden in der Regel in einem Semester abgeschlossen.

Modulgruppen sind Zusammenfassungen von Modulen mit einem weiteren inhaltlichen Zusammenhang. In allen Fällen stellt ein Modul oder ein Teilmodul eine Einheit dar, für die innerhalb und am Ende eines Semesters eine Prüfungsleistung erbracht werden kann, für die Leistungspunkte vergeben werden.

Wahlpflichtmodule werden bedarfsorientiert, meist in jährlichem Rhythmus angeboten. Das jeweilige Semester (Sommer- oder Wintersemester) kann der Modulbeschreibung entnommen werden. Grundsätzlich können Wahlpflichtmodule ab einer Untergrenze von 15 angemeldeten Teilnehmern durchgeführt werden. In besonderen Ausnahmefällen, z. B. bei wiederholter Unterschreitung der Mindestteilnehmerzahl oder erstmaligem Angebot einer Veranstaltung, kann die Fakultät von dieser Regelung abweichen. Die Entscheidung treffen der Fakultätsrat und der Dekan der Fakultät. Wahlpflichtmodule, die gleichzeitig in einem anderen Studiengang als Pflichtmodul gehalten werden, können auch stattfinden, wenn weniger als 15 Anmeldungen vorliegen.

Die Lehrveranstaltungen werden derzeit in deutscher Sprache gehalten.

Studienbegleitende Prüfungen und Studienfortschritt

Sämtliche Prüfungen erfolgen über das gesamte Studium verteilt studienbegleitend und stehen in direktem Bezug zur Lehrveranstaltung. Prüfungsbestandteile können je nach Lehrveranstaltung begleitend oder nach Abschluss des Moduls stattfinden, beispielsweise als Referat, Klausurarbeit, mündliche Prüfung, Hausarbeit mit Kolloquium, Entwurf mit Kolloquium, Laborbericht, Exkursionsbericht oder einer Kombination. In den Beschreibungen der einzelnen Module wird im Modulhandbuch die jeweilige Prüfungsform festgelegt. Es gelten allgemeine Studienfortschrittsberechtigungen, die die jeweils gültige Studien- und Prüfungsordnung regelt.

Zur Masterarbeit kann sich anmelden, wer mindestens 35 ECTS-Punkte erreicht und in mindestens 4 Pflichtfächern die Note "ausreichend" oder besser erzielt hat.

3 Modulbeschreibungen

IFM1101 Effiziente Algorithmen

Allgemeines

Modulverantwortliche(r):	Prof. Dr. Ulrich Göhner
Dozent(en):	Prof. Dr. Ulrich Göhner / Prof. Dr. Jochen Staudacher
Modultyp:	Pflichtmodul
Voraussetzungen:	Keine
Verwendbarkeit:	Studiengang Informatik (M.Sc.)
Angebot und Dauer:	Sommersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Mündliche Prüfung 30 Minuten am Ende des Semesters.
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

- Die Studierenden kennen die Begriffe der Graphentheorie
- Sie können wichtige Algorithmen zur Lösung graphentheoretischer Probleme anwenden und die Komplexität der Algorithmen und Methoden bewerten
- Sie können die gelernten Methoden auf praktische Beispiele anwenden
- Sie kennen die zentralen Methoden der Komplexitätstheorie und haben Verständnis für die Relevanz komplexitätstheoretischer Betrachtungen in der Praxis
- Sie können einfache Spiele klassifizieren und kennen den Nutzen spieltheoretischer Ansätze und Methoden in der Informatik
- Sie kennen Methoden und spieltheoretische Modelle zur Beschreibung von Auktionsverfahren und können diese auf einfache Fragestellungen aus der Internet-Ökonomie anwenden
- Sie haben Einblick in Funktionsweise, Grenzen und Möglichkeiten randomisierter Algorithmen
- Sie haben Einblick in Funktionsweise, Grenzen und Möglichkeiten des Prinzips der dynamischen Programmierung

Lehrinhalte:

- Begriffe und Definitionen der Graphen-Theorie
- Zusammenhang

- Topologisches Sortieren
- Algorithmen zur Bestimmung des kürzesten Wegs (A*, Dijkstra-, Floyd- und Warshall-Algorithmus)
- Lokale Suchalgorithmen (z.B. Bergsteiger- und Ameisen-Algorithmus)
- Bestimmung des minimalen Spannbaums mit Algorithmen von Prim und Kruskal
- Bestimmung des maximalen Flusses in Netzwerken mit den Algorithmen von Ford/Fulkerson, Edmonds/Karp und Dinic
- Lösung des Maximum-Matching Problems mit Hilfe von Netzwerken
- Praktischer Einsatz von Algorithmen auf Graphen mit BGL Boost
- Die Komplexitätsklassen P und NP
- NP-vollständige Probleme, insbesondere auch aus der Graphentheorie
- Grundelemente der Algorithmischen Spieltheorie
- Stabile Matchings, Mechanismen, Auktionen und deren praktische Anwendung in der Internet-Ökonomie
- Komplexität von Spielen
- Randomisierte Algorithmen
- Prinzipien der dynamischen Programmierung

Literatur:

- Logofatu, D.: "Algorithmen und Problemlösungen mit C++", Vieweg
- Heun, V.: "Grundlegende Algorithmen", Vieweg.
- Saake, G; Sattler, U.: "Algorithmen und Datenstrukturen", Dpunkt-Verlag
- Krumke, S, Noltemeier, H.: "Graphentheoretische Konzepte und Algorithmen", Teubner
- Gelperin, D.: "On the optimality of A* Artificial Intelligence, 8:69-76, 1977.
- Sedgewick, R. und J. S. Vitter: "Shortest paths in Euclidean graphs." Algorithmica, 1(1):31-48, 1986.
- S. Arora, B. Barak: Computational Complexity, Cambridge University Press, 1. Auflage, 2009.
- J.E. Hopcroft, R. Motwani, J.D. Ullman: Einführung in Automatentheorie, Formale Sprachen und Berechenbarkeit, Pearson Studium, 3. Auflage, 2011.
- M. Mitzenmacher, E. Upfal: Probability and Computing, Cambridge University Press, 2. Auflage, 2017.
- M.J. Osborne, A. Rubinstein: A Course in Game Theory, MIT Press, 1. Auflage, 1994.
- H. Peters: Game Theory, Springer, 2. Auflage, 2015.
- M. Maschler, E. Solan, S. Zamir: Game Theory, Cambridge University Press, 1. Auflage, 2013.
- Y. Saez, A. Mochon: Understanding Auctions, Springer, 1. Auflage, 2015.
- S.R. Chakravarty, M. Mitra, P. Sarkar: A Course on Cooperative Game Theory, Cambridge University Press, 2015

IFM1102 Mathematik

Allgemeines

Modulverantwortliche(r):	Prof. Dr. Christian Blick
Dozent(en):	Prof. Dr. Christian Blick / Epple
Modultyp:	Pflichtmodul
Voraussetzungen:	Analysis einer Veränderlichen, Lineare Algebra
Verwendbarkeit:	Studiengang Informatik (M.Sc.)
Angebot und Dauer:	Sommersemester, ein Semester
Lehrformen:	3 SWS Seminaristischer Unterricht 1 SWS Übung
Leistungspunkte:	5
Arbeitsaufwand:	60 Stunden Präsenzzeit Unterricht/Übung 30 Stunden Selbststudium - Betreute Studierzeit 60 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Schriftliche Prüfung 90 Minuten am Ende des Semesters
Zur Prüfung zugelassene Hilfsmittel:	kein Taschenrechner, ansonsten ohne Einschränkung

Lernergebnisse und Inhalte

Lernergebnisse:

Nach erfolgreicher Teilnahme am Modul Mathematik können die Studierenden

- zentrale Konzepte der Theorie linearer und nichtlinearer gewöhnlicher Differentialgleichungen wiedergeben
- komplexe Aufgabenstellungen hinsichtlich der Lösbarkeit mit Methoden der Theorie dynamischer Systeme analysieren
- geeignete Lösungsmethoden aus verschiedenen Teilgebieten der Mathematik auswählen und zu einem Lösungsansatz für Aufgaben aus dem Bereich dynamischer Systeme kombinieren

Lehrinhalte:

- Komplexe Zahlen, algebraische Operationen und deren geometrische Interpretation
- Komplexwertige Funktionen, Kurven in der komplexen Ebene, Grenzwerte, Differentiation und Integration komplexwertiger Funktionen
- Gewöhnliche lineare Differentialgleichungen; Homogene und inhomogene Gleichungen; Lösungsbegriff, reelle und komplexe Lösungen, Existenz und Eindeutigkeit; Lineare Differentialoperatoren, Lösungsraum, charakteristisches Polynom, Basisfunktionen; Partikuläre Lösung und Lösungen inhomogener Gleichungen; Differentialgleichungssysteme; Phasendiagramm und qualitative Eigenschaften der Lösungen
- Gewöhnliche nichtlineare Differentialgleichungen Beispiele; Phasendiagramm; Linearisierung; Stationäre, stabile und instabile Lösungen
- Fourier-Reihen, Fourier-Transformation; Definition, Konvergenz, Eigenschaften, DFT, FFT, Bandbegrenzte Funktionen, Anwendungsbeispiele

- Weiterer Anwendungsbereich

Literatur:

- Butz, T.: Fourier-Transformation für Fußgänger, B.G. Teubner, 2. Auflage
- Hachenberger, D.: Mathematik für Informatiker, Pearson Studium, 2. Auflage, 2008
- Hirsch, M.W., Smale, S.: Differential Equations, Dynamical Systems and Linear Algebra Academic Press, 1974
- Lang, H. W.: Algorithmen in Java, Oldenbourg Verlag, 2. Auflage, 2006
- Stingl, P.: Mathematik für Fachhochschulen, Hanser Verlag, 8. Auflage, 2009

IFM1103 Softwarearchitektur

Allgemeines

Modulverantwortliche(r):	Prof. Dr. Georg Hagel
Dozent(en):	Prof. Dr. Georg Hagel
Modultyp:	Pflichtmodul
Voraussetzungen:	keine
Verwendbarkeit:	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
Angebot und Dauer:	Sommersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Leistungsnachweis in der Übung, Schriftliche Prüfung 90 Minuten am Ende des Semesters. Leistungsnachweise sind Zulassungsvoraussetzungen.
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

Nach erfolgreicher Beendigung der Veranstaltung sind die Studierenden in der Lage,

- Zu erklären, welche Faktoren Einfluss auf die Architektur einer Software haben.
- Zu beschreiben, welche Architekturprinzipien beim Entwurf einer Softwarearchitektur beachtet werden müssen.
- Bekannte Architekturmuster bei der Erstellung der Softwarearchitektur anzuwenden.
- Aspekte von Softwaresystemen zu beschreiben.
- Lösungsansätze für Aspekte von Softwaresystemen anzugeben.
- Softwarearchitekturen mit aktuellen Beschreibungsmethoden zu dokumentieren.
- Continuous Integration, Continuous Deployment und DevOps zu beschreiben und anzuwenden.

Lehrinhalte:

- Aufgaben von Softwarearchitekten
- Entwurf von Softwarearchitekturen
- Architekturmuster
- Architektur Aspekte
- Dokumentation von Softwarearchitekturen
- Continuous Integration, Continuous Deployment und DevOps

Literatur:

- Richards, Mark, Ford, Neil: Handbuch moderner Softwarearchitektur, O'Reilly, 2021
- Starke, Gernot: Effektive Software-Architekturen, Carl Hanser Verlag, 9. Auflage (2020)
- Bass, Len et al.: Software Architecture in Practice, Fourth Edition, SEI Series in Software Engineering, Addison-Wesley (2021)
- Martin, Robert, C.: Clean Architecture, Prentice Hall, 2018
- Richardson, Chris: Microservices Patterns, Manning, 2019
- Oestereich, Bernd: Die UML 2.5 Kurzreferenz für die Praxis, Oldenbourg Verlag, 6. Auflage (2014)
- Buschmann, Frank et al.: POSA Band 1 - Pattern-orientierte Softwarearchitektur - Ein Pattern-System

IFM5101 3D Computer Vision

Allgemeines

Das Modul 3D Computer Vision behandelt ausgewählte Methoden der Projektiven Geometrie, der Posenbestimmung von Objekten und der 3D-Punktwolken-Analyse.

Es werden mathematischen Grundlagen zur Lösung linearer und nichtlinearer Gleichungssysteme behandelt und im Bereich der 3D Computer Vision angewendet. Dabei spielen homogene Koordinaten und homogene Gleichungssysteme eine entscheidende Rolle. Es werden Verfahren zur Kalibration von Kamerasystemen (Single View und Multiple View) behandelt und Methoden zur Triangulation auf der Basis der "Structure From Motion" vorgestellt.

Modulverantwortliche(r):	Prof. Dr. Klaus Ulhaas
Dozent(en):	Prof. Dr. Klaus Ulhaas
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	
Verwendbarkeit:	Studiengang Master Künstliche Intelligenz und Computer Vision, Master Informatik, Master Game-Engineering und Visual Computing
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übungen/Praktikum in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Schriftliche Prüfung 90 Minuten am Ende des Semesters
Zur Prüfung zugelassene Hilfsmittel:	nicht programmierbarer Taschenrechner

Lernergebnisse und Inhalte

Lernergebnisse:

Die Studierenden sind in der Lage

- 3D Punkte auf mehrere Kamerabildflächen zu projizieren
- aus Bildpunkten mehrerer Kameras entsprechende 3D Punkte zu rekonstruieren
- ein Mehrkamerasystem zu Kalibrieren
- die Epipolargeometrie zum Matching zu nutzen
- Lineare homogene Gleichungssysteme zu nutzen um unbekannte Posen-, Structure- oder Kameraparameter zu lösen
- metrische Rekonstruktionen durchzuführen

Lehrinhalte:

Das Modul 3D Computer Vision behandelt folgende Themen:

- Projektive Geometrie

- Homographie
- Kamerakalibration
- Epipolargeometrie: Essential und Fundamentalmatrix
- Triangulation und metrische Rekonstruktion
- Lineare und Nichtlineare Optimierung
- Bündelblock-Methoden (Bundle Adjustment)
- Pose Estimation
- Iterative Closest Point (ICP)
- Structure From Motion

Literatur:

Die Literatur wird zu den einzelnen Themen auf den Folien bekanntgegeben. Folgende Bücher sind zu empfehlen:

- 3D Point Cloud Analysis, Liu, Shan, Zhang, Min, Kadam, Pranav and Kuo, C. -C. Jay, Springer International Publishing, 2021
- Multiple View Geometry in Computer Vision, Hartley, Richard and Zisserman, Andrew, Cambridge University Press, Second Edition, 2003

IFM5102 Deep Learning für Computer Vision

Allgemeines

In diesem Modul werden Deep Learning Modelle und Techniken für die Verarbeitung von Bildern vermittelt. Im Fokus hierbei steht das Convolutional Neural Network (CNN), aber auch Varianten und neuere Ansätze werden vorgestellt. CNNs stellen heute den "Kern" vieler State-of-the-Art (SOTA) Computer Vision Algorithmen dar.

Wir starten das Modul mit einer kurzen Geschichte der Bildverarbeitung. Die Herausforderungen der Bildverarbeitung und wie diese über viele Jahrzehnte hinweg systematisch unterschätzt worden sind, werden aufgeführt. Danach wird das CNN eingeführt, das heute das "Arbeitspferd" der Deep Learning basierten Bildverarbeitung darstellt und sehr oft als sog. "Feature Extractor" oder "Backbone" für viele neuere Bildverarbeitungsmodelle verwendet wird. Um das CNN Basismodell richtig zu durchdringen werden wir die einzelnen Komponenten eines CNNs (Filterschichten, Poolingschichten, Klassifikatorschicht) genauer untersuchen und die Studierenden sollen anhand von Experimenten in den Übungen mit Varianten dieser Schichten arbeiten, um eine Intuition für die Bedeutung im Rahmen der gesamten Verarbeitung eines Bildes zu entwickeln.

Ein weiterer Bereich den wir verstehen wollen ist die Anwendung der Transformer-Architektur im Bereich Computer Vision: die sogenannten Vision-Transformer.

Meist werden heute Bildverarbeitungsmodelle nicht mehr "from-scratch" trainiert, sondern man nutzt bereits vortrainierte Modelle und passt diese jeweils auf die eigene konkrete Bildverarbeitungsaufgabe an. Wir schauen uns daher diesen Bereich des sog. "Finetunings" bzw. "Transferlernens" ebenfalls sehr genau an.

Es wird dann im Modul auch auf weiterführende Modelle eingegangen, die das CNN für Bildaufgaben im Kern benutzen, beispielsweise Objektdetektionsverfahren, Anomalie-Detektionsverfahren und Modelle für den Bereich Robotik.

Modulverantwortliche(r):	Prof. Dr. Jürgen Brauer
Dozent(en):	Prof. Dr. Jürgen Brauer
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	Das Modul Deep Learning ist explizit keine (!) Voraussetzung um dieses Modul hören / belegen zu können
Verwendbarkeit:	
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	4 SWS Seminaristischer Unterricht
Leistungspunkte:	5
Arbeitsaufwand:	60 Stunden Präsenzzeit Unterricht mit Praxisbeispielen 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Schriftliche Prüfung 90 Minuten am Ende des Semesters.
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

- Die Studierenden kennen die Herausforderungen bei der Bildverarbeitung, können diese durch konkrete Beispiele benennen.
- Die Teilnehmer des Moduls haben einen Überblick über die Geschichte der Bildverarbeitung bis hin zu Deep Learning basierten Modellen der Bildverarbeitung.
- Die Studierenden verstehen die Bedeutung der einzelnen Schichten eines Convolutional Neural Networks und wie die verschiedenen Schichtenarten bei der Bildklassifikation zusammenspielen.
- Die Teilnehmer des Moduls kennen aktuelle Modelle zur Objektdetektion und wissen wie diese sich unterscheiden. Sie verstehen die interne Arbeitsweise und können Vor- und Nachteile der einzelnen Ansätze benennen.
- Die Studierenden kennen Modelle zur Erzeugung realistisch aussehender Bilder und können deren Funktionsweise nachvollziehen.
- Die Studierenden können einzelne ausgewählte Deep Learning Verfahren zur Bildverarbeitung mit aktuellen Deep Learning Bibliotheken umsetzen.
- Studierende kennen den Aufbau und Funktionsweise der Vision Transformer und wissen wie man auch diese Architektur für Bildverarbeitungsprobleme nutzen kann.

Lehrinhalte:

- Computer Vision - Eine kurzer geschichtlicher Rückblick
- CNN - Gesamtüberblick über das Convolutional Neural Network
- CNN - Bedeutung und Funktionsweise der einzelnen Komponenten
- Deep Learning Modelle zur Objektdetektion
- Neuere Ansätze für die Feature Extraktion wie Vision Transformer
- Generative Deep Learning Modelle um realistisch aussehende Bilder zu erzeugen
- Vision Transformer - Die neue Generation der Bildverarbeitungsmodelle

Literatur:

IFM5103 Large Language Models

Allgemeines

Seit Einführung von ChatGPT erfreuen sich große Sprachmodelle großer Beliebtheit mit großem Anwendungspotential im privaten wie im industriellen Umfeld. Zugrunde liegt ein autoregressives Transformermodell, das statistische Gesetzmäßigkeiten auf großen Textbeständen lernt und reproduziert. Die Vorlesung führt ein in die mathematischen und technischen Grundlagen und zeigt mögliche Anwendungsfelder auf. Es werden Architekturelemente mit Python-Code nachvollzogen, daher werden Grundkenntnisse in Mathematik und Machine Learning vorausgesetzt und eine Vertrautheit mit Python und Jupyter Notebook empfohlen. Das Modul wird angeboten von Dr. Fabio Lischka und Christoph Wald, wissenschaftliche Mitarbeiter am Forschungszentrum Allgäu.

Modulverantwortliche(r):	Prof. Dr. Frank Schirmeier
Dozent(en):	Prof. Dr. Frank Schirmeier / N.N. Lehrbeauftragter / N.N. Lehrbeauftragter
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	
Verwendbarkeit:	
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Leistungsnachweise in den Übungen, Schriftliche Prüfung 90 Minuten am Ende des Semesters. Leistungsnachweise sind Zulassungsvoraussetzung.
Zur Prüfung zugelassene Hilfsmittel:	handschriftliche Notizen, 1 DIN A4 Blatt, beidseitig beschrieben, keine Kopie

Lernergebnisse und Inhalte

Lernergebnisse:

Die Studierenden sind in der Lage, unterschiedliche Architekturen von Large Language Models auf mathematische Formeln zu reduzieren und zu vergleichen. Sie sind in der Lage, ein einfaches Large Language Model in Python zu implementieren, Textdaten für das Training vorzuverarbeiten, und ein solches Training durchzuführen. Die Studenten erkennen Probleme in der Qualität der Trainingsdaten und können Gegenmaßnahmen zu ergreifen. Sie sind in der Lage, die grundlegenden mathematischen Modellannahmen zu erklären, auf denen Large Language Models aufbauen, und daraus Vorhersagen über deren Verhalten abzuleiten. Sie können LLMs in größere Anwendungen integrieren und wenden grundlegende Prinzipien bei der Einbettung von LLMs (LLMOps) an.

Lehrinhalte:

Inhalte zu nachfolgenden Themenfeldern werden bearbeitet:

- Natural Language Processing
- Architekturelemente von LLMs, insb. Transformer-Architektur
- Bereinigung von Textdaten
- Trainingsstrategien und Hyperparameter-Tuning für LLMs
- Instruction Tuning
- Prinzipien zur Anwendung von LLMs ("LLMOps")

Literatur:

Das Modul stützt sich neben aktuellen Veröffentlichungen u.a. auf nachfolgende Literatur:

- Chaumond et al.: LLM Engineer's Handbook
- Gazit et al.: Mastering NLP from Foundations to LLMs
- Raschka: Build a Large Language Model (from Scratch)
- Alto: Building LLM Powered Applications Create Intelligent Apps and Agents with Large Language Models

IFM5104 Reinforcement Learning

Allgemeines

Das Modul vermittelt die theoretischen Grundlagen und praktischen Methoden des Reinforcement Learning (RL), einer zentralen Teildisziplin des maschinellen Lernens, in der lernende Agenten durch Interaktion mit ihrer Umgebung optimale Handlungsstrategien entwickeln. Zu Beginn werden die konzeptionellen und mathematischen Grundlagen eingeführt, insbesondere Markov-Entscheidungsprozesse, Belohnungsfunktionen, Zustands- und Aktionsräume sowie das zentrale Spannungsfeld zwischen Exploration und Exploitation. Darauf aufbauend werden klassische Ansätze des Reinforcement Learning behandelt, darunter Dynamic Programming, Monte-Carlo-Methoden sowie Temporal-Difference-Verfahren wie SARSA und Q-Learning. Im weiteren Verlauf des Moduls wird der Übergang zum Deep Reinforcement Learning (DRL) vollzogen, bei dem neuronale Netze als Funktionsapproximatoren eingesetzt werden, um komplexe Zustandsräume zu bewältigen. Behandelt werden zentrale DRL-Algorithmen wie Deep Q-Networks (DQN) und deren Erweiterungen (Double DQN, Dueling DQN), Policy-Gradient-Methoden (REINFORCE, A2C/A3C) sowie moderne Verfahren wie Proximal Policy Optimization (PPO). Ein besonderer Fokus liegt auf den Herausforderungen bei der Stabilisierung des Lernprozesses, der effizienten Nutzung von Erfahrungen durch Replay Buffers sowie auf der Balance zwischen Lernstabilität und Exploration. Abschließend werden ausgewählte Anwendungen und aktuelle Forschungsthemen vorgestellt, unter anderem der Einsatz von RL in Spielen (z. B. Atari, Go, AlphaZero), in der Robotik, in autonomen Systemen sowie im Multi-Agent Reinforcement Learning. Darüber hinaus werden aktuelle Herausforderungen wie Sample Efficiency, Sicherheit, Interpretierbarkeit und ethische Fragestellungen diskutiert. Begleitend zur Vorlesung vertiefen die Studierenden die Inhalte in praktischen Übungen, in denen sie beispielsweise RL-Agenten in Simulationsumgebungen (z. B. Gymnasium) implementieren, trainieren und evaluieren.

Modulverantwortliche(r):	Prof. Dr. Jürgen Brauer
Dozent(en):	Prof. Dr. Jürgen Brauer
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	
Verwendbarkeit:	
Angebot und Dauer:	Sommersemester, ein Semester
Lehrformen:	4 SWS Seminaristischer Unterricht
Leistungspunkte:	5
Arbeitsaufwand:	60 Stunden Präsenzzeit Unterricht mit Praxisbeispielen 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Schriftliche Prüfung 90 Minuten am Ende des Semesters
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

- Die Studierenden erklären die grundlegenden Konzepte des Reinforcement Learning und ordnen es als Teildisziplin des maschinellen Lernens ein.

- Die Studierenden beschreiben Markov-Entscheidungsprozesse einschließlich Zustandsräumen, Aktionsräumen, Belohnungsfunktionen und Übergangsdynamiken.
- Die Studierenden erläutern das Spannungsfeld zwischen Exploration und Exploitation und dessen Bedeutung für den Lernprozess.
- Die Studierenden wenden klassische Reinforcement-Learning-Verfahren wie Dynamic Programming, Monte-Carlo-Methoden und Temporal-Difference-Lernen an.
- Die Studierenden erklären die Funktionsweise zentraler TD-Algorithmen wie SARSA und Q-Learning.
- Die Studierenden beschreiben die Motivation und Grundlagen des Deep Reinforcement Learning unter Verwendung neuronaler Netze als Funktionsapproximatoren.
- Die Studierenden erläutern Aufbau und Funktionsweise von Deep-Q-Networks sowie deren Erweiterungen wie Double DQN und Dueling DQN.
- Die Studierenden erklären Policy-Gradient-Methoden einschließlich REINFORCE, A2C/A3C und Proximal Policy Optimization.
- Die Studierenden analysieren Herausforderungen der Lernstabilisierung, der Experience-Replay-Nutzung und der Balance zwischen Stabilität und Exploration.
- Die Studierenden beurteilen ausgewählte Anwendungen von Reinforcement Learning in Spielen, Robotik, autonomen Systemen und Multi-Agenten-Szenarien.
- Die Studierenden diskutieren aktuelle Forschungsthemen und Herausforderungen wie Sample Efficiency, Sicherheit, Interpretierbarkeit und ethische Fragestellungen.
- Die Studierenden implementieren, trainieren und evaluieren Reinforcement-Learning-Agenten in Simulationsumgebungen anhand praktischer Übungen.

Lehrinhalte:

Literatur:

- Sutton, R. S. & Barto, A. G. (2018): Reinforcement Learning: An Introduction (2nd Edition). MIT Press.
- Graesser, L. & Keng, Wah Loon (2020): Foundations of Deep Reinforcement Learning. Pearson / Addison Wesley Data & Analytics Series

IFM5105 Physical AI

Allgemeines

Physical AI bezeichnet KI-Systeme, die die physische Welt wahrnehmen, Entscheidungen treffen und in ihr handeln – von Industrierobotern über humanoide Roboter bis zum autonomen Fahrzeug. Das Modul vermittelt den vollständigen Entwicklungszyklus solcher Systeme entlang des roten Fadens „Von der digitalen Szene über Simulation und Training zur realen, sicheren Maschine“: Die Studierenden modellieren Roboterzellen als Digitale Zwillinge mit OpenUSD, erzeugen synthetische Bilddaten für das Training von Perzeptionsmodellen, programmieren Roboter und trainieren lernbasierte Steuerungen per Imitation Learning, Reinforcement Learning und Robot Foundation Models. Eine Fallstudie überträgt die Konzepte auf das autonome Fahren; Sicherheitsarchitekturen, Normen und Regulierung (EU AI Act) schließen den Bogen zur verantwortungsvollen Praxis.

Modulverantwortliche(r):	Prof. Dr. Bernd Lüdemann-Ravit
Dozent(en):	Prof. Dr. Bernd Lüdemann-Ravit
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	Grundlagen Machine Learning / Deep Learning
Verwendbarkeit:	
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	2 SWS Seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Schriftliche Prüfung 90 Minuten am Ende des Semesters
Zur Prüfung zugelassene Hilfsmittel:	Skript (Ausdruck mit eigenen Notizen), nicht programmierbarer Taschenrechner

Lernergebnisse und Inhalte

Lernergebnisse:

Die Studierenden können nach erfolgreichem Abschluss des Moduls:

- Die Konzepte von Physical AI (Simulation-Training-Deployment, Sim-to-Real-Gap, Digital Twins) erklären und gegen klassische Robotik abgrenzen.
- Aufbau und Kompositionsmechanismen von OpenUSD sowie die Architektur von ROS 2 beschreiben.
- Die Paradigmen Imitation Learning, Reinforcement Learning und Robot Foundation Models (VLA, World Models) einordnen und ihre Einsatzgebiete gegeneinander abwägen.
- 3D-Szenen und Roboterzellen mit OpenUSD programmatisch modellieren und komponieren.
- Synthetische Bilddatensätze mit Domain Randomization erzeugen und damit Perzeptionsmodelle trainieren.

- Lernbasierte Steuerungen aus Demonstrationen bzw. per RL trainieren und auf verwandte Domänen (autonomes Fahren) übertragen.

Lehrinhalte:

Die folgende Liste gibt eine Übersicht der behandelten Themen

- Grundlagen und Weltmodellierung
- Synthetische Daten und Perzeption
- ROS und Roboterprogrammierung
- Lernende Systeme und Sim-to-Real
- Autonomes Fahren und Sicherheit

Literatur:

Lynch, Park: Modern Robotics – Mechanics, Planning, and Control (frei verfügbar)

- Szeliski: Computer Vision – Algorithms and Applications (frei verfügbar)
- NVIDIA Learn-OpenUSD-Kurse (docs.nvidia.com/learn-openusd); OpenUSD-Dokumentation (openusd.org)
- ROS-2-Tutorials (docs.ros.org); MoveIt-2-Dokumentation
- Ausgewählte Paper: ACT/ALOHA (Zhao et al.), Diffusion Policy (Chi et al.), SayCan (Ahn et al.), Code as Policies (Liang et al.), Domain Randomization (Tobin et al.)
- CARLA-Dokumentation und CARLA Leaderboard; nuScenes- und Waymo-Open-Data-set-Paper
- Überblickstexte zu ISO 10218/TS 15066, ISO 26262/SOTIF, EU AI Act und EU-Maschinenverordnung

IFM5301 Betriebssystem im Eigenbau

Allgemeines

Modulverantwortliche(r):	Prof. Dr. Stefan Frenz
Dozent(en):	Prof. Dr. Stefan Frenz
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	keine
Verwendbarkeit:	Studiengang Informatik (M.Sc.)
Angebot und Dauer:	Sommersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übung
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Leistungsnachweise zu den Übungen, mündliche Prüfung 30 Minuten am Ende des Semesters, Leistungsnachweise sind Zulassungsvoraussetzungen
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

- Die Studierenden verstehen die grundlegenden Herausforderungen an der Implementierung von Betriebssystemen.
- Sie kennen die im Umfeld von aktuellen 32 und auch 64 Bit Rechnern relevanten praktischen sowie die erforderlichen theoretischen Hintergründe.
- Sie verstehen das Zusammenspiel zwischen Compiler, Laufzeitumgebung und Betriebssystem.

Lehrinhalte:

- x86-Architektur in Bezug zur Betriebssystementwicklung
- Design-Kriterien für maßgeschneiderte Betriebssysteme
- Gerätetreiber im allgemeinen und speziellen
- Anforderungen an Laufzeitsysteme
- Compiler-Grundlagen

Literatur:

- Tanenbaum, Andrew S.: Moderne Betriebssysteme (3. Auflage, 2009)
- Wirth, Niklaus; Gutknecht, Jürg: Project Oberon (2005)

IFM5302 Echtzeitsysteme

Allgemeines

Das Modul führt in das Thema Echtzeitsysteme ein und gibt einen Überblick über spezifische Problemstellungen und Lösungen im Echtzeitbetrieb. Dies beinhaltet neben Aspekten mit Bezug zu Echtzeitbetriebssystemen (RTOS) auch solche aus dem Bereich der Echtzeitkommunikation.

Modulverantwortliche(r):	Prof. Dr. Tim Poguntke
Dozent(en):	Prof. Dr. Tim Poguntke
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	
Verwendbarkeit:	Studiengang Informatik (M.Sc.)
Angebot und Dauer:	Sommersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	Vorlesung: 15 x 2 h = 30,0 h; Praktikum, Übung: 15 x 2 h = 30,0 h; Selbststudium: 90,0 h; Gesamtaufwand: 150,0 h
Leistungsnachweis und Prüfung:	Portfolioprüfung
Zur Prüfung zugelassene Hilfsmittel:	handschriftliche Notizen, 1 DIN A4 Blatt, beidseitig beschrieben, keine Kopie, nicht programmierbarer Taschenrechner

Lernergebnisse und Inhalte

Lernergebnisse:

Nach erfolgreicher Beendigung der Veranstaltung sind die Studierenden in der Lage,

- Statische und messbasierte Analysen der maximalen Ausführungszeiten von Codeabschnitten durchzuführen
- kooperative und präemptive MultitaskingVerfahren für Echtzeitsysteme zu evaluieren und anhand von Beispielen zu illustrieren,
- das Problem der Prioritätsinversion zu diskutieren und die Prioritätsvererbung zu beschreiben und darzustellen
- Methoden für die Zugriffskontrolle von Betriebsmitteln zu erläutern und illustrieren,
- eine Anwendung mit vernetzten Sensorknoten und Echtzeitkommunikation, z.B. für den Austausch von Telemetriedaten, zu entwickeln.

Lehrinhalte:

- Definition von Echtzeit
- Analyse maximaler Ausführungszeiten (WCET)
- Multitasking und Scheduling
- Echtzeitbetriebssysteme (RTOS)
- Präemptive und kooperative Scheduling-Verfahren (z.B. EDF, LLF, RMS)

- Semaphoren und Mutexe
- Inversion und Vererbung von Prozessprioritäten
- Ausgesuchte Aspekte aus der Echtzeitkommunikation

Literatur:

Wird in der Veranstaltung bekannt gegeben.

- Begleitendes und zusätzliches Lernmaterial ist im Hochschulnetz zugänglich.

IFM5303 Multimodale Sensorsysteme

Allgemeines

Die Lehrveranstaltung befasst sich mit der Verarbeitung von Sensordaten - insbesondere der Fusion von Sensordaten, die aus mehreren (auch unterschiedlichen) Sensordatenquellen stammen. Dies können zum Beispiel mehrere Kamerabilder sein, die zu einer Objektdetektion fusioniert werden. Als grundlegende Techniken werden vor allem der rekursive Bayessche Schätzer als etablierter Zustandsschätzer sowie lernende Fusionsarchitekturen, wie das Convolutional Neural Network, behandelt.

Modulverantwortliche(r):	Prof. Dr. Jürgen Brauer
Dozent(en):	Prof. Dr. Jürgen Brauer
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	Lineare Algebra, Programmierkenntnisse
Verwendbarkeit:	Studiengang Informatik (M.Sc.)
Angebot und Dauer:	Sommersemester, ein Semester
Lehrformen:	4 SWS Seminaristischer Unterricht
Leistungspunkte:	5
Arbeitsaufwand:	60 Stunden Präsenzzeit Unterricht mit Praxisbeispielen 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Schriftliche Prüfung 90 Minuten am Ende des Semesters
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

- Die Studierenden wissen wieso die Sensordatenfusion in realen multi-modalen Sensorsystemen notwendig ist und können Fusionsansätze klassifizieren
- Sie kennen und verstehen den probabilistischen Hintergrund des Zustandsschätzproblems
- Sie kennen eine konkrete Realisierung des rekursiven Bayesschen Schätzer durch den Partikel-Filter
- Sie kennen und verstehen neuere Architekturen zur Sensordatenfusion wie Convolutional Neural Networks (CNNs) und wie diese für konkrete Aufgaben wie die Objektdetektion eingesetzt werden können

Lehrinhalte:

- Warum ist Sensordatenfusion wichtig?
- Klassifikation von Fusionsansätzen
- Probabilistischer Rahmen des Zustandsschätzproblems: der rekursive Bayessche Filter
- Realisierung des rekursiven Bayesschen Schätzers über Partikel-Filter
- Repräsentation von Sensordaten in hierarchischen Fusionsarchitekturen wie CNNs zur Objektdetektion

- Representation Learning

Literatur:

- Francois Chollet. Deep Learning with Python. Manning, 2018.

IFM5304 Mobile Roboter

Allgemeines

Mobile Roboter sind immer noch nicht in unserem Alltag allgegenwärtig. Zwar gibt es einige wenige Beispiele wie Staubsaugroboter, Drohnen, Marserkundungsroboter, autonome Fahrzeuge oder Fußballspielende Roboter beim RoboCup, doch meist handelt es sich bisher um Nischenlösungen. Wieso ist es so schwer mobile Roboter zu bauen, die Arbeiten übernehmen können, die heute nur von Menschen ausgeführt werden können? In der Vorlesung soll ein breiter Überblick über die verschiedenen Aspekte des Themas mobile Roboter erarbeitet werden. Nachdem ein Verständnis für die vielseitigen Herausforderungen der mobilen Robotik geschaffen wird, werden wir auf geeignete Sensorik und Aktorik für Roboter eingehen. Die Organisation der Informationsverarbeitung ist in einem mobilen Roboter ein wichtiges zentrales Thema. Entsprechend werden hier verschiedene Ansätze für sog. Roboterkontrollarchitekturen vorgestellt. Ein ebenso zentrales Thema ist das Thema der Navigation: wie kann ein mobiler Roboter robust durch eine bekannte oder sogar unbekannte Umgebung navigieren? Schließlich behandeln wir das Thema des Lernens von geeigneten Aktionen für Roboter. Hier haben sich verschiedene Ansätze wie z.B. das Reinforcement-Learning und das Imitationslernen etabliert. Am Ende der Vorlesung werden wir auf Besonderheiten der kooperativen Robotik eingehen und es werden Ansätze vorgestellt, die eine Zusammenarbeit von mehreren Robotern ermöglichen sollen. In den Übungen sollen zentrale Algorithmen anhand von Robotersimulatoren vertieft werden.

Modulverantwortliche(r):	Prof. Dr. Jürgen Brauer
Dozent(en):	Prof. Dr. Jürgen Brauer
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	Analysis, Lineare Algebra, Programmierkenntnisse
Verwendbarkeit:	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
Angebot und Dauer:	Sommersemester, ein Semester
Lehrformen:	4 SWS Seminaristischer Unterricht
Leistungspunkte:	5
Arbeitsaufwand:	60 Stunden Präsenzzeit Unterricht mit Praxisbeispielen 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Schriftliche Prüfung 90 Minuten am Ende des Semesters
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

- Die Studierenden können einschätzen, welche Fähigkeiten heute von mobilen Robotern erwartet werden können und welche nicht
- Sie wissen welche Sensorik für eine robuste Umfelderkennung notwendig ist und kennen Ansätze zur Umfelderkennung auf Basis geeigneter Sensordaten
- Sie haben fundiertes Wissen über verschiedene Ansätze von Roboterkontrollarchitekturen und kennen deren Vor- und Nachteile

- Die Studierenden kennen unterschiedliche Navigationsverfahren und können diese selber implementieren und anwenden
- Sie haben einen breiten Überblick über Verfahren die das Erlernen von Aktionen erlauben, kennen den theoretischen Hintergrund der Verfahren und können die Verfahren selber umsetzen
- Den Studierenden sind mögliche Auswirkungen der Verfügbarkeit intelligenter mobiler Roboter auf die Gesellschaft bewusst und können kritisch beurteilt werden

Lehrinhalte:

- Einführung: Geschichte der mobilen Roboter
- Herausforderungen für mobile Roboter
- Roboterkontrollarchitekturen
- Navigation, Lokalisation und Erstellung von Karten (SLAM), Pfadplanung
- Lernen von Aktionen, Reinforcement-Learning, Imitationslernen
- Kognitive Architekturen für Roboter
- Ausblick: wohin entwickelt sich der Bereich mobile Roboter?
- Mögliche Auswirkungen der Verfügbarkeit intelligenter mobiler Roboter auf die Gesellschaft

Literatur:

- Joachim Hertzberg et al. Mobile Roboter: Eine Einführung aus Sicht der Informatik Springer Vieweg Verlag. 2012.
- Roland Siegwart et al. Introduction to Autonomous Mobile Robots MIT. 2011

IFM5309 Kryptographie

Allgemeines

Wir beschäftigen uns mit verschiedenen gängigen Kryptographieverfahren (siehe Themenliste) sowie grundsätzlichen Fragestellungen rund um die Kryptographie: Woher kommt die Kryptographie? Was ist nötig um Kryptographie zu betreiben? Wann ist sie sicher? Wie wirkt sich Kryptographie (bzw. ihr Abwesenheit) auf die moderne Gesellschaft aus? Wie können Wahlen sicher durchgeführt werden? Wie können Kryptosystem analysiert/gebrochen werden? Welche Entwicklungen, die die Kryptographie wesentlich verändern werden, sind mittelfristig zu erwarten (Stichwort Quantencomputer)? Was gibt es für Ansätze um dem entgegenzuwirken?

Modulverantwortliche(r):	Prof. Dr. Elmar Böhler
Dozent(en):	Prof. Dr. Elmar Böhler
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	Programmieren I+II, Algorithmen und Datenstrukturen, Theoretische Informatik, Diskrete Mathematik, Wahrscheinlichkeitsrechnung
Verwendbarkeit:	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Vorlesung 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Teilnahmepflicht in den Übungen, Schriftliche Prüfung 90 Minuten am Ende des Semesters. Erfolgreiche Teilnahme an den Übungen ist Zulassungsvoraussetzung.
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

Lehrinhalte:

Es werden die Grundlagen der Kryptographie behandelt, sowie einige der wesentlichsten verwendeten Verfahren. Mögliche Themen sind die folgenden:

- Historische Kryptosysteme und ihre Analyse
- Blockchiffren
- Betriebsarten
- DES
- AES

- Schlüsselaustausch
- Public-Key Kryptographie
- RSA
- Digitale Signaturen
- Primzahltests
- ElGamal
- Einführung in ECC
- Teilen von Geheimnissen
- Einblicke in die Quantenkryptographie
- Postquantenkryptographie
- Hashfunktionen
- Wahlverfahren
- Blockchains
- Pseudozufallszahlen
- Kryptographie in der Gesellschaft
- Zero-Knowledge Systeme
- Homomorphe Verschlüsselung
- Komplexitätstheoretische Grundlagen
- Berechnungsverifikation

Literatur:

- J. Buchmann: Einführung in die Kryptographie
- J. Rothe: Complexity Theory and Cryptology
- K. Wagner: Theoretische Informatik
- A. Menezes, P. van Oorschot, S. Vanstone: Handbook of Applied Cryptography

IFM5311 Computer Vision

Allgemeines

Modulverantwortliche(r):	Prof. Dr. Klaus Ulhaas
Dozent(en):	Prof. Dr. Klaus Ulhaas
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	
Verwendbarkeit:	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Schriftliche Prüfung 90 Minuten am Ende des Semesters
Zur Prüfung zugelassene Hilfsmittel:	nicht programmierbarer Taschenrechner

Lernergebnisse und Inhalte

Lernergebnisse:

Die Studierenden sind in der Lage

- den theoretischen Hintergrund klassischer Verfahren und moderner Verfahren zu erklären.
- die gelernten Verfahren zu kombinieren und mittels Pytorch und OpenCV auf Beispiele aus der Praxis zur Objekterkennung und -segmentierung zu übertragen.
- mathematische Zusammenhänge zu verstehen und anzuwenden.

Lehrinhalte:

Folgende Inhalte werden in der Veranstaltung bearbeitet:

- Einführung in Python, OpenCV und Pytorch
- Klassische 2D Computer Vision Verfahren wie z.B. HOG- und SIFT-Detektor
- Basisoperationen für Tiefe Neuronale Netze (Faltung, Max-Pooling, Max-Unpooling, Up-Convolution)
- Convolutional Neural Networks
- Segmentierung (klassisch: Selective Search & modern: U-Net)
- Objektlokalisierung und Objektdetektion
- Moderne Objektdetektoren: R-CNN, Fast-RCNN, Faster R-CNN, YOLO
- Implementierung von klassischen Computer Vision Verfahren mittels OpenCV und modernen Computer Vision Verfahren mittels Pytorch
- Metriken zur Objektdetektion wie z.B. Precision, Recall, F β -Maß, Konfusionsmatrix,

mAP, Multiclass-Confusion-Matrix

Literatur:

Literatur und Empfehlungen zu Einzelthemen werden fortlaufend in der Vorlesung bekanntgegeben

IFM5402 Identity- und IT-Architekturmanagement

Allgemeines

Modulverantwortliche(r):	Prof. Dr. Peter Klutke
Dozent(en):	Prof. Dr. Peter Klutke
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	keine
Verwendbarkeit:	Studiengang Informatik (M.Sc.)
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Praktikum in kleinen Projektteams
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Vorlesung 30 Stunden Präsenzzeit Praktikum 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Mündliche Prüfung 20 Minuten am Ende des Semesters
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

Nach erfolgreicher Beendigung der Veranstaltung sind die Studierenden in der Lage, ...

- die Grundlagen des Identitymanagements und des IT-Architekturmanagements strukturiert darzustellen und zu erklären,
- dazugehörige Konzepte und Strategien zu vergleichen, zu beurteilen und auf neue Situationen anzuwenden und
- gesellschaftliche Auswirkungen und Sozialverträglichkeit von Lösungen und Innovationen besser zu beurteilen, etwa im Identity Management bei der Authentifizierung und Autorisierung in Notsituationen oder im Enterprise Architecture Management beim mehrdimensionalen Denken betreffend das "Enterprise Intergrating" und die "Enterprise Ecological Adaptation".

Lehrinhalte:

- Grundlagen des Identitymanagements mit Einführung, Begriffen, einem Ebenenmodell, zugehörigen Prozessen und Technologien, Rollenmodellen sowie organisationsübergreifendem Identitymanagement (Federation)
- Grundlagen des IT-Architekturmanagements mit Einführung, Begriffen, Bedeutung, Bestandteilen, der Architekturpyramide und deren Schichten, Modellen zur Architekturplanung, Referenzarchitekturen sowie dazugehörigen Prozessen
- Besprechung von Fallbeispielen zur Veranschaulichung im Praktikum unter Einsatz aktivierender Lernmethoden.

Literatur:

- Alexander Tsolkas, Klaus Schmidt: "Rollen und Berechtigungskonzepte: Identity- und Access-Management im Unternehmen", Verlag: Springer Vieweg; Auflage: 2. Aufl. 2017 (14. Juli 2017), ISBN-13: 978-3658179861

- Christian Mezler-Andelberg: „Identity Management – eine Einführung“; dpunkt.verlag; 1. Aufl. (Okt. 2007)
- Inge Hanschke: "Enterprise Architecture Management - einfach und effektiv: Ein praktischer Leitfaden für die Einführung von EAM", Verlag: Carl Hanser Verlag GmbH & Co. KG; Auflage: 2., überarbeitete (8. August 2016), ISBN-13: 978-3446447240
- Gernot Dern: "Management von IT-Architekturen: Leitlinien für die Ausrichtung, Planung und Gestaltung von Informationssystemen"; Vieweg+Teubner Verlag; 3., durchges. Aufl. (März 2009)

IFM5405 Integration und Management von Geschäftsprozessen mit SAP

Allgemeines

Modulverantwortliche(r):	Prof. Dr. Stefan Wind
Dozent(en):	Prof. Dr. Stefan Wind
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	Keine
Verwendbarkeit:	Studiengang Informatik (M.Sc.)
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Die erfolgreiche Teilnahme an der Übung gilt als Zulassungsvoraussetzung für die 90 minütige schriftliche Prüfung am Ende des Semesters.
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

Die Studierenden

- kennen moderne Integrationsplattformen und Integrationsszenarien
- sind in der Lage, komplexe, integrierte Geschäftsprozesse zu analysieren, zu beschreiben und zu modellieren
- sind durch ein grundsätzliches Verständnis befähigt, integrierte Informationssysteme am Beispiel von SAP ERP Systemen zu verstehen und zu beschreiben
- können diese Kenntnisse auf die Integration und das Management der grundlegenden Geschäftsprozesse in den Bereichen Kundenauftragsmanagement, Beschaffung, Finanzwesen usw. entlang der horizontalen und vertikalen Integration in SAP ERP-Systemen transferieren

Lehrinhalte:

- Grundlagen der Integration und Management von Geschäftsprozessen mit SAP
- Moderne Integrationsplattformen und Szenarien (z. B. SAP NetWeaver)
- Integrierte Informationssysteme am Beispiel von ERP und In-Memory-basierte ERP-Systemen von SAP
- Modellierung von integrierten Geschäftsprozessen und Geschäftsprozessmanagement mit ARIS
- Integrierte Geschäftsprozesse und prozessorientierte Integration der betrieblichen Informationsverarbeitung in SAP

- Fallstudien zu komplexen integrierten Geschäftsprozessen und Informationssystemen mit SAP ERP GBI

Literatur:

- Mertens, P.: Integrierte Informationsverarbeitung 1 - Administrations- und Dispositionssysteme in der Industrie, 18. Aufl., Wiesbaden 2013
- Mertens, P., Meier M.: Integrierte Informationsverarbeitung 2 - Planungs- und Kontrollsysteme in der Industrie, 10. Aufl., Wiesbaden 2009
- Scheer, A.-W.: Wirtschaftsinformatik - Referenzmodelle für industrielle Geschäftsprozesse, Studienausgabe, 2. Aufl., Berlin, Heidelberg, New York 1998
- Scheer, August-Wilhelm. ARIS-vom Geschäftsprozess zum Anwendungssystem. Springer-Verlag, 2013

IFM5406 Funktionale Sicherheit

Allgemeines

Einführung in Funktionale Sicherheit mit besonderer Beachtung der Anforderungen von Informatik, Fahrerassistenzsystemen und Automatisierung & Robotik.

Modulverantwortliche(r):	Prof. Dr. Rolf Jung
Dozent(en):	Prof. Dr. Rolf Jung
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	keine
Verwendbarkeit:	Studiengang Informatik (M.Sc.)
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	Übung/Praktikum findet während des Semesters in Blöcken statt.
Leistungspunkte:	5
Arbeitsaufwand:	45 Stunden Präsenzzeit Unterricht 15 Stunden Präsenzzeit Praktikum/Übung 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Schriftliche Prüfung 90 Minuten am Ende des Semesters
Zur Prüfung zugelassene Hilfsmittel:	Benutzung von Teilen der Normen ISO 26262 und ISO 13849 nach Vorgabe, Skript (Ausdruck mit eigenen Notizen), nicht programmierbarer Taschenrechner

Lernergebnisse und Inhalte

Lernergebnisse:

- Studenten kennen den Begriff 'Funktionale Sicherheit' und damit verbundene Definitionen aus den Normen
- Teilnehmer kennen die Elemente des Sicherheitsprozesses und können sie dem Entwicklungsprozess zuordnen
- Sie kennen Managementmethoden und können diese dem Sicherheitslebenszyklus zuordnen
- Studenten können die Methode der HARA anwenden
- Teilnehmer können zwischen reparierbaren und nicht reparierbaren Elementen unterscheiden und die Funktionen Unzuverlässigkeit und Nichtverfügbarkeit technischer Elemente erklären
- Studenten kennen die unterschiedlichen logischen Operatoren der Wahrscheinlichkeitskombinatorik und kennen die Modellbildung in der FTA bzw. anderen Sicherheitsanalysen
- Studenten können die Methode FMEDA im Sicherheitsprozess einordnen sowie Sicherheitskennzahlen bestimmen und anwenden
- Teilnehmer kennen Testmethoden bezüglich System, Hardware, Software und können aus System- und Produktanforderungen Testanforderungen erstellen
- Studenten können die Anforderungen der Norm ISO26262 zum Produktionsprozess

im Sicherheitslebenszyklus anwenden

- Sie kennen den Begriff der Zugangssicherheit im Kontext von Funktionaler Sicherheit
- Studenten können Softwarewerkzeuge klassifizieren
- Die Studenten kennen einen Zulassungsprozess und können Dokumente für eine Sicherheitszulassung zusammenstellen

Lehrinhalte:

- Einführung in die Gesetzeslage und Normenüberblick mit Begriffserklärungen
- Erläuterung der Elemente eines Sicherheitsprozesses und Erstellung eines Sicherheitsplans und Sicherheitsnachweises
- Analyse der Elemente und Schritte in einem Sicherheitslebenszyklus
- Durchführung einer Risiko- und Gefährdungsanalyse
- Ermittlung von Sicherheitskennzahlen an einem Fallbeispiel
- Einführung in mathematischen Methoden der Zuverlässigkeitstheorie und Durchführung einer Sicherheitsanalyse für Fehlerkombinationen
- Erarbeitung einer FMEDA mit Fallbeispiel
- Management der Funktionalen Sicherheit
- Kennenlernen von Testmethoden und Erstellen eines V&V-Plans
- Einführung in Zugangssicherheit, Cybersecurity in Verbindung mit Funktionaler Sicherheit
- Klassifizierung von Softwarewerkzeugen entsprechend Sicherheitsnormen

Literatur:

- Josef Börcsök, Funktionale Sicherheit: Grundzüge sicherheitstechnischer Systeme, VDE Verlag 2014
- Löw, Pabst, Petry: Funktionale Sicherheit in der Praxis, dpunkt.verlag
- Ross, Hans-Leo: Funktionale Sicherheit im Automobil, Hanser Verlag
- Gebhardt, Rieger, Mottok, Gießelbach: Funktionale Sicherheit nach ISO 26262, dpunkt.verlag
- Gebhardt, Rieger, Mottok, Gießelbach: Funktionale Sicherheit und Cybersecurity, dpunkt.verlag

IFM5407 E-Business Management

Allgemeines

Modulverantwortliche(r):	Prof. Dr. Stefan Wind
Dozent(en):	Prof. Dr. Stefan Wind
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	keine
Verwendbarkeit:	Studiengang Informatik (M.Sc.)
Angebot und Dauer:	Sommersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übungen/Praktikum in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung/Praktikum 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Schriftliche Prüfung 90 Minuten am Ende des Semesters
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

Die Studierenden

- sind in der Lage, die durch elektronische Technologien induzierten Veränderungen traditioneller Geschäftsprozesse zu erklären
- sind durch ein grundsätzliches Verständnis des E-Business befähigt, die Möglichkeiten innovativer Verfahren zur Information, Kommunikation und Transaktion zu beschreiben
- kennen elektronische Geschäftsprozesse und -modelle in der Net Economy
- können diese Kenntnisse auf elektronische Kontaktnetzwerke (E-Community), den elektronischen Handel (E-Marketplace), Systeme bei der elektronischen Kooperation (E-Company), Systeme im elektronischen Einkauf (E-Procurement) und Systeme im elektronischen Verkauf (E-Shop) transferieren

Lehrinhalte:

- Grundlagen des E-Business Managements
- Geschäftsprozessmanagement im E-Business
- Prozesse, Systeme, Management, Integration und Implementierungen in den Bereichen E-Procurement, E-Community, E-Company, E-Shop und E-Marketplace
- Neuen Trends und Technologien im E-Busienss

Literatur:

- Kollmann, Tobias. Das Digital Leadership. Springer Fachmedien Wiesbaden, 2020
- Kollmann, Tobias. Digital Business: Grundlagen von Geschäftsmodellen und-prozessen in der Digitalen Wirtschaft. Springer Gabler, 2022

-
- Kollmann, Tobias, ed. Handbuch Digitale Wirtschaft. Wiesbaden: Springer Gabler, 2020
 - Wirtz, Bernd. Digital Business: Strategien, Geschäftsmodelle und Technologien. Wiesbaden: Springer Fachmedien Wiesbaden, 2024

IFM5409 Requirements Engineering und Management

Allgemeines

Modulverantwortliche(r):	Prof. Dr. Georg Hagel
Dozent(en):	Prof. Dr. Georg Hagel
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	Keine
Verwendbarkeit:	Studiengang Master Informatik
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht - Just-in-Time-Teaching - Feedback durch Quizzes 2 SWS Übung und Vorträge in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Schriftliche Prüfung 90 Minuten am Ende des Semesters, Leistungsnachweise in der Veranstaltung, Leistungsnachweise sind Zulassungsvoraussetzungen.
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

Nach erfolgreicher Beendigung der Veranstaltung sind die Studierenden in der Lage,

- die Aufgaben von Requirements-Engineers zu beschreiben.
- Techniken, mit denen funktionale und nicht-funktionale Anforderungen erhoben und dokumentiert werden zu beschreiben.
- Techniken, mit denen funktionale und nicht-funktionale Anforderungen erhoben und dokumentiert werden, anzuwenden.
- qualitätssichernde Regeln des SOPHIST-Regelwerks auf bestehende Anforderungen anzuwenden.
- funktionale und nicht-funktionale Anforderungen für Softwaresysteme zu unterscheiden, und qualitätszusichern.
- durch die Lehr- und Lernform "Just-in-Time-Teaching" selbstständig Lerninhalte zu erarbeiten - eine für das lebenslange Lernen notwendige Technik.
- den Einsatz von KI im Requirements Engineering zu beurteilen.

Lehrinhalte:

Der Requirements-Engineer

- Anforderungsermittlung
- Das SOPHIST-Regelwerk
- Schablonen für Anforderungen

- Dokumentation von Anforderungen
- Nicht-funktionale Anforderungen

Literatur:

Chris Rupp et al.: Requirements-Engineering und -Management, Hanser, diverse Auflagen

- Christoph Ebert: Systematisches Requirements Engineering, dPunkt, diverse Auflagen
- Klaus Pohl: Requirements Engineering, Springer, 2. Aufl. 2025
- James Robertson et al.: Mastering the Requirements Process, Pearson, 4. Aufl. 2025
- Die SOPHISTen: KI im Requirements Engineering, dpunkt, 2025

IFM5601 Deep Learning

Allgemeines

Modulverantwortliche(r):	Prof. Dr. Jürgen Brauer
Dozent(en):	Prof. Dr. Jürgen Brauer
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	Analysis, Lineare Algebra, Programmierkenntnisse
Verwendbarkeit:	Studiengang Informatik (M.Sc.), Studiengang Künstliche Intelligenz und Computer Vision (M.Sc.)
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	4 SWS Seminaristischer Unterricht
Leistungspunkte:	5
Arbeitsaufwand:	60 Stunden Präsenzzeit Unterricht mit Praxisbeispielen 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Schriftliche Prüfung 90 Minuten am Ende des Semesters.
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

- Die Studierenden kennen mindestens drei der wichtigsten DL Architekturen im Detail
- Für jeden dieser drei Ansätze haben Sie fundiertes Wissen über den Aufbau und Funktionsweise gesammelt und können dieses Wissen bei Problemstellungen im Bereich Klassifikation, Regression und Spracherkennung durch Anwendung eines geeigneten DL Verfahrens umsetzen.
- Die Studierenden verstehen, welche Eigenschaften die DL Verfahren besonders auszeichnet und von klassischen Machine Learning Verfahren unterscheidet
- Sie kennen die historische Entwicklung des DL Gebietes und können abschätzen welche weiteren Entwicklungen zu erwarten sind
- Die Studierenden kennen die Herkunft vieler Ideen des Deep Learnings, die oft auf Erkenntnisse aus dem Bereich Neurowissenschaften zurück gehen

Lehrinhalte:

- Motivation Deep Learning: kürzliche Erfolge
- Geschichte des Deep Learnings
- Wieso boomt Deep Learning erst jetzt?
- Das biologische Neuron als Vorbild für das technische Neuronenmodell
- Technische Neuronenmodelle
- Ein Neuronales Netz ohne Deep Learning Bibliothek selber implementieren
- Perzeptron / Perzeptron-Lernregel
- Multi-Layer-Perzeptron

- Backpropagation-Algorithmus
- Deep Learning Bibliotheken
- Rekurrente Neuronale Netze
- Deep Learning zur Textverarbeitung
- Die Transformer-Architektur
- Optimierer und Deep Learning Tricks
- Ausblick: wohin entwickelt sich der Ansatz?

Literatur:

- Yann LeCun, Yoshua Bengio, Geoffrey Hinton. Deep Learning. Nature, 2015.
- Jürgen Schmidhuber. Deep Learning in Neural Networks: An Overview. Neural Networks Journal 61 (2015): 85-117
- Dive into Deep Learning: <https://d2l.ai/>

IFM5603 Datenvisualisierung

Allgemeines

Die Lehrveranstaltung vermittelt Techniken und Algorithmen der wissenschaftlichen Datenvisualisierung von großen Datenmengen. Weiterhin werden verschiedene Anwendungsfelder und Methoden der Datenerhebung behandelt.

Modulverantwortliche(r):	Prof. Dr. Christoph Bichlmeier
Dozent(en):	Prof. Dr. Christoph Bichlmeier
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	Grundlegende Kenntnisse in der Computergraphik sowie in der C++ Programmierung
Verwendbarkeit:	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
Angebot und Dauer:	Sommersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übungen (verpflichtend)
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übungen/Praktikum 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Erfolgreiche Teilnahme an den Übungen, schriftl. Prüfung 90 Minuten am Ende des Semesters
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

Nach erfolgreicher Teilnahme an der Lehrveranstaltung kennen die Studierenden verschiedene Algorithmen, Methoden und Anwendungsbereiche wissenschaftlichen Datenerhebung und Datenvisualisierung. Sie können für einige diese Anwendungsbereiche selbstständig Algorithmen implementieren.

Lehrinhalte:

Die Studierenden können mithilfe von Excel, C++, Qt, OpenGL und GLSL Lösungen für folgende Themenbereiche realisieren.

- Wissenschaftliche Durchführung einer Datenerhebung und Datenauswertung
- Visuelle Darstellung von statistischen Datenerhebungen
- Visualisierung von Volumendaten und medizinischen Schnittbildern
- Visualisierung von Strömungsdaten
- Visualisierung von Netzwerkdaten
- Interpolationsverfahren für die Datenvisualisierung
- Methoden der Datenerhebung in verschiedenen Anwendungsfeldern

Literatur:

Literaturliste

-
- Visual Computing for Medicine: Theory, Algorithms, and Applications; Bernhard Preim and Charl P. Botha
 - Visualisierung: Grundlagen und allgemeine Methoden; 2013; Heidrun Schumann, Wolfgang Müller
 - Diverse Paper aus verschiedenen Quellen (Fachzeitschriften und Konferenzbände)

IFM5604 Parallelprogrammierung

Allgemeines

Modulverantwortliche(r):	Prof. Dr. Ulrich Göhner
Dozent(en):	Prof. Dr. Ulrich Göhner
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	Keine
Verwendbarkeit:	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übung in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Schriftliche Prüfung 90 Minuten am Ende des Semesters
Zur Prüfung zugelassene Hilfsmittel:	handschriftliche Notizen, 1 DIN A4 Blatt, beidseitig beschrieben, keine Kopie

Lernergebnisse und Inhalte

Lernergebnisse:

- Die Studierenden kennen aktuelle parallele Rechnerarchitekturen einordnen und deren Vor- und Nachteile
- Sie verstehen die wichtigsten parallele Programmierkonzepte
- Sie können die Leistung von Parallelrechnern bei der Lösung praktischer Probleme bewerten
- Sie kennen die Vorgehensweisen bei der effektiven Parallelisierung von Algorithmen
- Sie beherrschen den effektiven Einsatz moderner paralleler Programmierwerkzeuge

Lehrinhalte:

- Einführung
- Leistungsbewertung, Amdahl und Gustafson-Gesetz
- parallele Rechnerarchitekturen
- Programmierkonzepte für Parallelrechner
- OpenMP und MPI
- GPU-Programmierung mit CUDA und OpenCL
- PRAM (Parallel Random Access Machine)
- Parallele Algorithmen und Komplexität
- Präfixberechnungen
- Parallele Berechnung numerischer Ausdrücke

- Parallele Matrix- und Sortieralgorithmen
- Grid- und Cloud-Computing
- Quantenrechner

Literatur:

- Tanenbaum, A.S.: "Computerarchitektur. Strukturen - Konzepte - Grundlagen", Pearson Studium, 2006.
- "OpenMP Application Program Interface", Version 4.0 - RC 1, November 2012.
- "MPI: A Message-Passing Interface Standard", Version 3.0, Message Passing Interface Forum, September 21, 2012.
- "CUDA C PROGRAMMING GUIDE", NVIDIA, 2012.
- "Introduction to OpenCL Programming", AMD, 2010.
- Homeister, M.: "Quantum Computing verstehen", Springer Fachmedien Wiesbaden, 2015.
- Balaji, P.: "Compute Unified Device Architecture," Programming Models for Parallel Computing , MITP, 2015.
- Rauber, Th.; Rünger, G.: "Parallele Programmierung", Springer Vieweg Berlin Heidelberg, 2012.
- Rahm, E.; Sattler, K.-U.: " Verteiltes und Paralleles Datenmanagement", Springer Vieweg Berlin Heidelberg, 2015.

IFM5605 Algorithmen und Strategien zur Entscheidungsunterstützung

Allgemeines

Modulverantwortliche(r):	Prof. Dr. Jochen Staudacher
Dozent(en):	Prof. Dr. Jochen Staudacher
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	Keine
Verwendbarkeit:	Studiengang Informatik (M.Sc.), Studiengang Game Engineering und Visual Computing (M.Sc.)
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	3 SWS Seminaristischer Unterricht 1 SWS Praktikum in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	45 Stunden Präsenzzeit Vorlesung 15 Stunden Präsenzzeit Praktikum 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Die erfolgreiche Teilnahme an den Übungen gilt als Zulassungsvoraussetzung für eine 20-minütige mündliche Prüfung am Ende des Semesters.
Zur Prüfung zugelassene Hilfsmittel:	keine Hilfsmittel

Lernergebnisse und Inhalte

Lernergebnisse:

- Die Studierenden können entscheidungs- und spieltheoretische Konzepte auf praktische Fragestellungen aus Informatik und Wirtschaftswissenschaften anwenden.
- Sie können beschreiben und analysieren, welchen Einfluss die Struktur eines sozialen oder ökonomischen Netzwerks auf strategisches Handeln besitzt.
- Sie können für ausgewählte Anwendungsfälle aus großen Datensätzen sinnvolle Information extrahieren, die zur Entscheidungsunterstützung eingesetzt werden kann.
- Sie können die vorgestellten Methoden und Algorithmen in R umsetzen.
- Sie können zu den vorgestellten Ansätzen und Algorithmen zur Entscheidungsunterstützung kritisch Stellung nehmen und deren Potenziale und Grenzen charakterisieren.

Lehrinhalte:

- Entscheidungskriterien, Entscheidungsmodelle, Entscheidungsalgorithmen
- Nutzenfunktionen, rationales Entscheiden bei Risiko, Bernoulli-Prinzip
- Interaktives Entscheiden, Algorithmische Spieltheorie und Anwendungen in Informatik und Ökonomie
- Anpassung und Erweiterung spieltheoretischer Ansätze zur Analyse ökonomischer und sozialer Netzwerke

- Ökonomien mit Netzwerkeffekten
- Statistische Entscheidungstheorie
- Datenbasiertes Entscheiden: Algorithmen und Anforderungen an die Datenqualität

Literatur:

Es gibt nicht das EINE Lehrbuch zu dieser Vorlesung; verschiedene Teile der Vorlesung werden durch die folgenden Lehrbücher abgedeckt:

- James N. Webb: Game Theory: Decisions, Interaction and Evolution, Springer, 2007
- Michael Maschler, Eilon Solan, Shmuel Zamir: Game Theory, Cambridge University Press, 2013
- Hans-Jürgen Zimmermann: Operations Research, Vieweg, 2. Auflage, 2008
- Matthew O. Jackson: Social and Economic Networks, Princeton University Press, 2008
- David Easley, Jon Kleinberg: Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010
- Pierre Lafaye de Micheaux, Remy Drouilhet, Benoit Liqueur: The R Software, Springer, 2014
- Larry Wasserman: All of statistics, Springer, 2004

IFM5606 Big Data

Allgemeines

Die Datenmengen, die in unserer Kommunikations- und Wissensgesellschaft anfallen wachsen unaufhörlich und die Datenvielfalt nimmt zu. Sie bieten ein riesiges Potential um mehr über uns, unsere Gesellschaft, unsere Umwelt und unsere Wirtschaft zu erfahren. Daten müssen jedoch zunächst einmal gespeichert, verarbeitet und analysiert werden. Dies stellt ganz neue Anforderungen an die zugrundeliegenden Systeme, da die klassischen Ansätze mit relationalen Datenbanken für strukturierte Daten hierfür nicht mehr geeignet sind. In Big Data Systemen werden die Daten verteilt auf viele Knoten gespeichert und verarbeitet. Die Daten selbst sind oft semistrukturiert oder unstrukturiert und müssen ggf. erst aufbereitet und gefiltert werden. Analysemethoden werden immer ausgefeilter und zunehmend ist eine Analyse in Echtzeit gefordert, die wiederum neue Technologien und Systeme erfordert.

In der Vorlesung zu Big Data werden verschiedene Problemstellungen und Lösungsansätze für Big Data Systeme diskutiert. Im Praktikum werden diese Ansätze mit einigen der derzeit leistungsfähigsten Systeme auf reale Daten angewendet.

Modulverantwortliche(r):	Prof. Nikolaus Steger
Dozent(en):	Prof. Nikolaus Steger
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	Keine
Verwendbarkeit:	Studiengang Informatik (M.Sc.)
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Praktikum in kleinen Gruppen
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Vorlesung 30 Stunden Präsenzzeit Praktikum 90 Selbststudium
Leistungsnachweis und Prüfung:	Aktive Teilnahme am Praktikum. Schriftliche Prüfung 90 Minuten am Ende des Semesters. Erfolgreiche Teilnahme am Praktikum ist Zulassungsvoraussetzung für die Prüfung.
Zur Prüfung zugelassene Hilfsmittel:	handschriftliche Notizen, 1 DIN A4 Blatt, beidseitig beschrieben, keine Kopie

Lernergebnisse und Inhalte

Lernergebnisse:

Nach erfolgreicher Beendigung der Veranstaltung sind die Studierenden in der Lage

- die Funktionsweise, die theoretischen Grundlagen und wichtige Algorithmen und Datenstrukturen zur Speicherung, Analyse und verteilten Verarbeitung großer Datenmengen zu verstehen
- konkrete Komponenten für Big Data Systeme zu kennen und in der Praxis einzusetzen
- komplexe Big Data Systeme zu konzipieren und aufzubauen

Lehrinhalte:

- Einsatzbereiche von Big Data Systemen
- Map-Reduce-Paradigma, Apache Hadoop, HDFS
- Batchverarbeitung mit Apache Spark, Resilient Distributed Datasets, Data Frames und Spark SQL
- Data Warehousing
- SQL-Erweiterungen für Analytics und Data Warehousing
- Column Store, Datenkomprimierung und Datenstrukturen zur Speicherung analytischer Daten
- Online Analytics und Stream Processing mit Apache Flink und Flink SQL
- Anforderungen, allgemeine Konzepte und Komponenten für Big Data Systeme

Literatur:

- Learning Spark, Lightning-Fast Data Analysis- Damji, Wenig, Das, O'Reilly and Associates

IFM5609 Data Science

Allgemeines

Data Science ist ein interdisziplinäres Wissenschaftsfeld mit Schnittmengen zur Informatik und zur Statistik und beschäftigt sich mit der Extraktion von Wissen aus Daten. Das Ziel ist es, Erkenntnisse und Handlungsempfehlungen in konkreten Problemstellungen zu generieren und zugleich, wo möglich, das Maß der Unsicherheit in den Erkenntnissen zu quantifizieren. Angesichts der sprunghaft wachsenden Datengenerierung in allen Bereichen der modernen Gesellschaft findet Data Science vielzählige kommerzielle Einsatzmöglichkeiten, beispielsweise im Kontext Industrie 4.0, insbesondere, aber nicht ausschließlich, in Big Data Szenarien. Die Veranstaltung geht auf Grundlagen der Statistik ein, erklärt grundlegende Methoden des Maschinellen Lernens, betont die Rolle der Datenvisualisierung und des Storytellings und versucht darüberhinaus das Verständnis für typische Fallstricke und allgemeiner für die Grenzen der Dateninterpretation zu schärfen. Mathematik-Kenntnisse im Umfang der Grundvorlesungen (Summationsformeln, Integration, einfache Beweise) werden vorausgesetzt. Der Umgang mit Python und gängigen Data Science Paketen in Verbindung mit Jupyter Notebooks wird in den Übungen erklärt und einstudiert.

Modulverantwortliche(r):	Prof. Dr. Frank Schirmeier
Dozent(en):	Prof. Dr. Frank Schirmeier
Modultyp:	Wahlpflichtmodul
Voraussetzungen:	
Verwendbarkeit:	
Angebot und Dauer:	Wintersemester, ein Semester
Lehrformen:	2 SWS Seminaristischer Unterricht 2 SWS Übung
Leistungspunkte:	5
Arbeitsaufwand:	30 Stunden Präsenzzeit Unterricht 30 Stunden Präsenzzeit Übung 90 Stunden Selbststudium
Leistungsnachweis und Prüfung:	Leistungsnachweise in den Übungen, Schriftliche Prüfung 90 Minuten am Ende des Semesters. Leistungsnachweise sind Zulassungsvoraussetzung.
Zur Prüfung zugelassene Hilfsmittel:	handschriftliche Notizen, 1 DIN A4 Blatt, beidseitig beschrieben, keine Kopie

Lernergebnisse und Inhalte

Lernergebnisse:

Die Studierenden sind in der Lage, in einem Unternehmen im Gespräch mit einem Mitarbeiter der Führungs- oder Fachebene eine datenbezogene Problemstellung zu analysieren, die relevanten Daten zu identifizieren, diese in einer geeigneten Infrastruktur zu erfassen, weiterzuverarbeiten und in nachfolgenden Analyseschritten Handlungsempfehlungen zu erarbeiten. Hierzu wird Python in Verbindung mit Jupyter Notebook eingesetzt. Es werden grundlegende Verfahren der Statistik und des Maschinellen Lernens beherrscht. Bei ungenügender Datenqualität oder -quantität sind die Studierenden befähigt, zu argumentieren, dass keine valide Aussage getroffen werden kann.

Lehrinhalte:

- Deskriptive Statistik (Mittelwert, Median, Pearson- und Spearman-Korrelation, Quantile, ...)
- Statistische Tests (u-Test, t-Test) und Bootstrapping
- Underfit und Overfit (Bias-Variance-Tradeoff)
- Grundlegende Methoden des Maschinellen Lernens (Dimensionsreduktion, Regression, Kernel Density-Estimation) und Anwendungen zur Signalanalyse und Anomalie-detektion
- Datenreinigung und Feature Engineering
- Projektmanagement im Data Science-Umfeld (insb. CRISP-DM)
- Python und Jupyter Notebook mit gängigen Paketen wie numpy, pandas, scikit-learn oder matplotlib

Literatur:

- Jake VanderPlas: Data Science Handbook (online verfügbar)
- Wes McKinney: Python for Data Analysis
- J. Grus: Einführung in Data Science
- S. Chan: Introduction to Probability for Data Science
- T. Hastie, R. Tibshirani, J. Friedman: The Elements of Statistical Learning
- G. James, D. Witten, T. Hastie, R. Tibshirani: An Introduction To Statistical Learning
- Christopher M. Bishop: Pattern Recognition and Machine Learning

IFM2000 Hauptseminar

Allgemeines

Modulverantwortliche(r):	Alle Professoren der Fakultät
Dozent(en):	Alle Professoren der Fakultät
Modultyp:	Pflichtmodul
Voraussetzungen:	Keine
Verwendbarkeit:	Studiengang Informatik (M.Sc.)
Angebot und Dauer:	Sommersemester, ein Semester
Lehrformen:	Seminar
Leistungspunkte:	5
Arbeitsaufwand:	15 Stunden Präsenzzeit Vorträge und anschließende Diskussion 135 Stunden Vorbereitung der Präsentation und Ausarbeitung der Studienarbeit
Leistungsnachweis und Prüfung:	Seminarvortrag mit anschließender Diskussion Studienarbeit

Zur Prüfung zugelassene Hilfsmittel:

Lernergebnisse und Inhalte

Lernergebnisse:

Nach erfolgreicher Beendigung der Veranstaltung sind die Studierenden in der Lage

- sich in ein anspruchsvolles, wissenschaftliches Thema einzuarbeiten
- entsprechende Literatur zu recherchieren und aufzuarbeiten
- einen Vortrag zu einem wissenschaftlichen Thema zu erarbeiten, zu präsentieren und bei der nachfolgenden Diskussion zu vertreten
- eine schriftliche Ausarbeitung zu einem wissenschaftlichen Thema nach den Grundsätzen guter wissenschaftlicher Praxis zu erstellen

Lehrinhalte:

Abhängig vom Thema des Hauptseminars

Literatur:

Abhängig vom Thema des Hauptseminars in Absprache mit dem Dozenten

- Balzert, H., Schröder, M., Schäfer, C.: Wissenschaftliches Arbeiten, W3L Verlag, 2. Auflage, 2012
- Wytrzens, Hans Karl: Wissenschaftliches Arbeiten : eine Einführung, 5., überarbeitete Auflage, Wien, ISBN: 9783708915005
- DFG: Sicherung guter wissenschaftlicher Praxis - Denkschrift, WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, ergänzte Auflage, 2013

IFM3000 Projekt

Allgemeines

Modulverantwortliche(r):	Alle Professoren der Fakultät
Dozent(en):	Alle Professoren der Fakultät
Modultyp:	Pflichtmodul
Voraussetzungen:	Keine
Verwendbarkeit:	Studiengang Informatik (M.Sc.)
Angebot und Dauer:	Sommersemester, ein Semester
Lehrformen:	Projekt
Leistungspunkte:	5
Arbeitsaufwand:	15 Stunden Präsenzzeit Besprechung mit betreuendem Professor 135 Stunden Durchführung des Projektes
Leistungsnachweis und Prüfung:	Studienarbeit Kolloquium
Zur Prüfung zugelassene Hilfsmittel:	

Lernergebnisse und Inhalte

Lernergebnisse:

Nach erfolgreicher Beendigung der Veranstaltung sind die Studierenden in der Lage

- ein Projekt zu planen, zu organisieren und zu leiten
- ein Projekt zu dokumentieren und gegenüber Dritten zu vertreten
- Probleme im Projekt frühzeitig zu erkennen und zu lösen
- ein Projekt erfolgreich abzuschließen
- das erworbene Wissen auch in komplexen Anwendungsszenarien einzusetzen und erfolgreich anzuwenden
- aus unvollständigen und ggf. widersprüchlichen Anforderungen in Abstimmung mit dem Auftraggeber konsistente und überzeugende Lösungen zu erarbeiten
- sich schnell in neue Anwendungsgebiete, Technologien und Grundlagen einzuarbeiten

Lehrinhalte:

Abhängig vom Projektauftrag in Absprache mit dem jeweiligen Dozenten

Literatur:

Abhängig vom Projektauftrag in Absprache mit dem jeweiligen Dozenten

IFM6000 Masterarbeit

Allgemeines

Modulverantwortliche(r):	Betreuender Professor
Dozent(en):	Betreuender Professor
Modultyp:	Pflichtmodul
Voraussetzungen:	Mindestens 35 CP und in mindestens 4 Pflichtmodulen die Note
Verwendbarkeit:	Studiengang Informatik (M.Sc.)
Angebot und Dauer:	Die Frist zur Bearbeitung der Masterarbeit im Vollzeitstudium beträgt 6 Monate, im Teilzeitstudium 12 Monate.
Lehrformen:	
Leistungspunkte:	30
Arbeitsaufwand:	900 Stunden
Leistungsnachweis und Prüfung:	Abschlussarbeit
Zur Prüfung zugelassene Hilfsmittel:	

Lernergebnisse und Inhalte

Lernergebnisse:

- Fähigkeit die im Studium erworbenen Kenntnisse und Fertigkeiten in einer selbständig angefertigten, anwendungsorientiert-wissenschaftlichen Arbeit auf komplexe Aufgabenstellungen anzuwenden
- Fähigkeit sich selbständig in ein wissenschaftliches Thema einzuarbeiten
- Fähigkeit aus unvollständigen und widersprüchlichen Informationen in Abstimmung mit dem Aufgabensteller eine konsistente und überzeugende Lösung zu erarbeiten
- Die Abschlussarbeit darf mit Zustimmung der Prüfungskommission in einer Einrichtung außerhalb der Hochschule ausgeführt werden.

Lehrinhalte:

Literatur:

In Absprache mit dem betreuenden Professor

- Wytrzens, Hans Karl: Wissenschaftliches Arbeiten : eine Einführung, 5., überarbeitete Auflage, Wien, ISBN: 9783708915005
- DFG: Sicherung guter wissenschaftlicher Praxis - Denkschrift, WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim, ergänzte Auflage, 2013